МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

На правах рукописи

Мукосей Анатолий Викторович

Алгоритмы выбора узлов и построения таблиц маршрутов для высокоскоростной сети с топологией «многомерный тор»

Специальность 2.3.5 —
«Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва — 2023

Работа выполнена в отделе архитектуры и программного обеспечения суперкомпьютеров АО «Научно-исследовательский центр электронной вычислительной техники».

Научный руководитель: **Семенов Александр Сергеевич**, кандидат технических наук

Официальные оппоненты: **Якобовский Михаил Владимирович,** доктор физико-математических наук, профессор, член-корреспондент РАН, Институт прикладной математики им. М.В. Келдыша РАН, заместитель директора

Посыпкин Михаил Анатольевич, доктор физико-математических наук, доцент, член-корреспондент РАН, Федеральный исследовательский центр «Информатика и управление» РАН, заместитель директора

Баранов Антон Викторич, кандидат технических наук, доцент, Межведомственный суперкомпьютерный центр РАН – филиал ФГУ «Федеральный научный центр Научно-исследовательский институт системных исследований РАН», заместитель директора

Защита состоится 22 декабря 2023 г. в 15:00 на заседании диссертационного совета МГУ.012.2 при Московском государственном университете имени М.В. Ломоносова по адресу: 119991, Москва, ГСП-1, Ленинские горы, МГУ, д. 1 строение 52, факультет ВМК, аудитория №238.

E-mail: ilgova@cs.msu.su.

С диссертацией можно ознакомиться в отделе диссертаций научной библиотеки МГУ имени М.В. Ломоносова (Ломоносовский просп., д. 27) и на сайте https://dissovet.msu.ru/dissertation/012.2/2767.

Автореферат разослан «___»_____ 2023 года.

Ученый секретарь
диссертационного совета
МГУ.012.2,
кандидат физико-математических
наук Антонов Александр Сергеевич

Общая характеристика работы

Актуальность

В настоящее время суперкомпьютеры имеют важное значение как для научных и промышленных приложений, так и для обеспечения обороноспособности страны. Производительность суперкомпьютеров растет высокими темпами за счет увеличения количества ядер и применения ускорителей в вычислительных узлах, а для обмена данными и синхронизации между узлами необходима высокоскоростная коммуникационная сеть. Зачастую именно коммуникационная сеть определяет реальную производительность, в особенности на задачах с интенсивным обменом данными в условиях распределенной вычислительной системы.

При создании коммуникационных сетей одной из распространенных топологий являются топологии типа «многомерный тор». Данные топологии используются в суперкомпьютерах IBM Blue Gene/Q, K computer/PRIMEHPC FX10/Fugaku, Sugon, при этом в таких сериях могут создаваться как уникальные суперкомпьютеры из первой десятки списка самых мощных суперкомпьютеров в мире Top500, так и небольшие системы для нужд промышленных организаций.

Сеть Ангара – первая российская высокоскоростная коммуникационная сеть на основе СБИС маршрутизатора. СБИС маршрутизатора коммуникационной сети является разработкой АО «НИЦЭВТ» и выпущен по технологии 65 нм. Сеть поддерживает топологию «многомерный тор» (возможны варианты от 1D- до 4D-тор). В настоящее время существует более десяти высокопроизводительных вычислительных систем от 8 до 92 узлов, использующих сеть Ангара.

Суперкомпьютер – это высокопроизводительная вычислительная система коллективного пользования, поэтому при эксплуатации суперкомпьютеров в условиях наличия отказов (каналов связи или узлов) и узлов, занятых заданиями других пользователей, необходимо предоставлять возможность выделения или выбора для задания пользователя достижимого множества узлов, отвечающего запрашиваемым вычислительным мощностям, а также создавать в этих множествах таблицы маршрутизации.

При этом необходимо учитывать для сетей с топологией «многомерный тор» специфические требования маршрутизации и возможности отказов, требования равномерности распределения (балансировки) сетевого трафика, минимизации фрагментации, минимизации числа возможных транзитных узлов. В силу противоречивости ряда требований важен выбор разумного компромисса. От качества решения задачи построения сбалансированных таблиц маршрутов зависит производительность при выполнении задания пользователя, а от расширения возможности выбора узлов зависит эффективность использования суперкомпьютера с точки зрения выполнения вычислительными узлами полезной работы.

Степень разработанности темы

Научно-технические решения по выбору узлов и построения таблиц маршрутов в коммуникационной сети с топологией «многомерный тор» в условиях наличия отказов и занятых узлов разработаны недостаточно. Основная причина – различные особенности маршрутизации в сетях с данной топологией создают специфические условия, в которых необходимо решать задачу построения таблиц маршрутов. Также при решении задачи выбора узлов для топологии «многомерный тор» часто используются подходы с избыточностью и необходимо оценивать утилизацию вычислительной системы во время запуска заданий пользователей. Для сети Ангара существовали базовые алгоритмы решения задач построения таблиц маршрутов и выбора узлов, однако они не пригодны к использованию в условиях возможных отказов, не используют все возможности маршрутизации и допускают потерю возможных решений задачи выбора узлов. Все перечисленные аргументы, а также возрастающие требования практики при использовании сети Ангара в научных и промышленных организациях определяют актуальность данной диссертационной работы.

Цель и задачи диссертационной работы

Целью работы является расширение возможности выбора множества узлов в сети Ангара в условиях наличия занятых и отказавших ресурсов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Разработать алгоритм для анализа маршрутов в сетях с топологией «многомерный тор» с учетом отказавших узлов и каналов связи и ограничений маршрутизации на маршрут сетевого пакета в зависимости от истории его прохождения по сети.
2. Разработать алгоритм определения достижимости множества вычислительных узлов.
3. Разработать алгоритм построения таблицы маршрутов для решения задачи балансировки трафика.
4. Разработать алгоритм выбора множества вычислительных узлов требуемого размера.
5. Реализовать и провести исследование разработанных алгоритмов по сравнению с существовавшими ранее алгоритмами.

Научная новизна

1. Впервые предложен алгоритм построения маршрутного графа для сетей с топологией «многомерный тор» и произвольным количеством отказавших узлов и каналов связи, а также маршрутизацией, накладывающей ограничение на маршрут сетевого пакета в зависимости от истории его прохождения по сети. Данный маршрутный граф позволяет применять алгоритмы обработки графов для анализа маршрутов в коммуникационной сети.
2. Впервые разработаны алгоритмы определения достижимости множества вычислительных узлов, построения таблиц маршрутов и выбора множества узлов для сети Ангара с возможными отказами узлов и каналов связи.
3. В отличие от существовавших ранее алгоритмов разработанные алгоритмы охватывают несколько уровней системного программного обеспечения суперкомпьютера и позволяют на уровне управления ресурсами суперкомпьютера учитывать аппаратные возможности маршрутизации сети Ангара.

Практическая ценность работы

1. Разработанные алгоритмы реализованы в программном дополнении (плагине) ANSU для системы Slurm управления заданиями на вычислительном кластере, которая является частью системного программного обеспечения сети Ангара.
2. Дополнение ANSU в составе системного программного обеспечения используется в семи вычислительных системах, построенных в различных организациях с использованием сети Ангара.
3. Внедрение разработанного дополнения ANSU позволило на суперкомпьютере «Десмос» в ОИВТ РАН повысить утилизацию вычислительных ресурсов на 7,65%.

Методология и методы исследования

При получении основных результатов диссертационной работы использовалась теория графов, методы анализа таблиц маршрутов, имитационное моделирование работы вычислительной системы с использованием синтетических очередей пользовательских заданий. При разработке реализаций предложенных алгоритмов и плагина ANSU для Slurm использовались методы объектно–ориентированного анализа и проектирования на языке C++, для имитационного моделирования – средства прототипирования с использованием языка Python.

Основные положения, выносимые на защиту

1. Разработан алгоритм построения маршрутного графа для анализа маршрутов в высокоскоростных коммуникационных сетях с топологией «многомерный тор» с произвольным количеством отказавших узлов и каналов связи, а также маршрутизацией, накладывающей ограничение на маршрут сетевого пакета в зависимости от истории его прохождения по сети. Временная сложность алгоритма $O(N^2)$, где N – количество узлов в сети.
2. Разработан алгоритм определения достижимости множества вычислительных узлов сети размера N, временная сложность алгоритма $O(N^2)$. Алгоритм использует возможность программного контроля отсутствия дедлоков в сети Ангара, что позволяет сохранять достижимость сети при большем числе случайно отказавших каналов связи (от 5% до 34%) по сравнению с возложением контроля отсутствия дедлоков на аппаратные возможности сети Ангара.
3. Разработан алгоритм построения таблицы маршрутов для решения задачи балансировки трафика в достижимом множестве узлов размера N, временная сложность алгоритма $O(N^2)$.
4. Разработан алгоритм выбора узлов в сети размера N с учетом её фрагментации, временная сложность алгоритма $O(N^4)$. Алгоритм позволил по сравнению с существовавшими ранее алгоритмами от 2 до 12 раз расширить возможности при выборе множества узлов в сети Ангара в зависимости от потока пользовательских заданий и исследуемой системы.
5. Разработанные алгоритмы реализованы и используются в составе системного программного обеспечения десяти вычислительных систем, построенных с использованием сети Ангара.

Степень достоверности и апробация результатов

Основные результаты работы докладывались на конференциях и семинарах:

1. Национальный Суперкомпьютерный Форум, 30 ноября – 3 декабря 2021.
2. Международная конференция «Суперкомпьютерные дни в России», 27–28 сентября 2021.
3. Международная конференция «Суперкомпьютерные дни в России», 23–24 сентября 2019.
4. Международная конференция «Суперкомпьютерные дни в России», 24–25 сентября, 2018.
5. XII Международная научная конференция «Параллельные вычислительные технологии» (ПаВТ'2018), 2–6 апреля 2018.

6. Школа-семинар «Поиск эффективных суперкомпьютерных архитектур в пост-Муровскую эру», МИЭМ НИУ ВШЭ, 11 декабря 2017.
7. Международная конференция «Суперкомпьютерные дни в России», 26–27 сентября 2016.
8. X Международная научная конференция «Параллельные вычислительные технологии» (ПаВТ'2016), 28 марта – 1 апреля 2016.
9. Научный семинар НИВЦ МГУ под руководством В.В. Воеводина.
10. Научный семинар МСЦ РАН под руководством Б.М. Шабанова.
11. Научный семинар ИПМ РАН под руководством М.В. Якобовского.
12. Научный семинар ЮУрГУ под руководством Л.Б. Соколинского.
13. Научный семинар кафедры ИИТ факультета ВМК МГУ под руководством И.В. Машечкина.
14. Научный семинар кафедры АСВК факультета ВМК МГУ под руководством Р.Л. Смелянского.

Личный вклад

Все представленные результаты в диссертационной работе получены лично автором. Подготовка части материалов к публикациям проводилась совместно с соавторами, причем вклад диссертанта был определяющим. В работах [1–3;5–10] А.С. Семенову, А.С. Симонову и Д.В Макагону принадлежит постановка задач и консультирование. В работе [1] А.А. Третьякову принадлежит выполнение оценочного тестирования производительности на суперкомпьютере, данные результаты не вошли в диссертационную работу. В работе [4] автору принадлежит реализация алгоритмов выбора узлов в программном дополнении Slurm.

Публикации

Основные положения и выводы диссертационного исследования в полной мере изложены в 11 научных работах [1–11]. 7 работ [1–7] опубликованы в рецензируемых научных изданиях, определенных п. 2.3 Положения о присуждении ученых степеней в Московском государственном университете имени М.В. Ломоносова, 4 из которых [1–4] изданы в журналах, индексируемых Scopus/Web of Science; а 3 другие работы [5–7] изданы в журналах, рекомендованных ВАК при Минобрнауки России.

Объем и структура работы

Диссертация состоит из введения, пяти глав и заключения. Полный объём диссертации составляет 147 страниц, включая 46 рисунков и 13 таблиц. Список литературы содержит 112 наименований.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится степень разработанности темы, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

Первая глава посвящена описанию высокоскоростной сети Ангара и применяемых в ней алгоритмов маршрутизации, а также обзору литературы по основным способам анализа маршрутов в сетях, алгоритмам построения таблиц маршрутов, а также алгоритмам выбора узлов в суперкомпьютерах.

Раздел 1.1 посвящен описанию высокоскоростных сетей, которые имеют топологию «многомерный тор». Топология типа «многомерный тор» представляет собой узлы, соединенные в многомерную решетку, крайние узлы которой соединены. Примером одномерной торовой топологии является кольцо.

К преимуществам топологии тор можно отнести масштабируемость; хорошая производительность на задачах физического 3D-моделирования, которые естественным образом накладываются на торовую топологию; избыточность соединений, что повышает отказоустойчивость и увеличивает бисекционную пропускную способность.

Высокоскоростные сети с топологией «многомерный тор» получили широкое распространение и используются в различных суперкомпьютерах. Примерами высокопроизводительных сетей в суперкомпьютерах на протяжении последних двадцати лет являются: Cray T3E, Cray Seastar, Cray Seastar2/2+, Cray Gemini, IBM Blue Gene L/P с топологией 3D-тор. Среди современных сетей можно выделить американскую сеть IBM Blue Gene Q с топологией 5D-тор, японскую сеть Tofu/Tofu2/Tofu-D с топологией 6D-тор, европейскую сеть Extoll с топологией 3D-тор, китайскую сеть Sugon с топологией 3D-тор.

В России также разрабатывались сети с топологией «многомерный тор»: СКИФ-Аврора и Паутина с топологией 3D-тор, СМПО-10G. Сеть Ангара – первая российская высокоскоростная коммуникационная сеть на основе СБИС маршрутизатора. СБИС маршрутизатора коммуникационной сети является российской разработкой и выпущен в 2013 году по технологии 65 нм. Сеть поддерживает топологию 4D-тор.

В *разделе 1.2* описываются основные принципы маршрутизации в сетях с топологией «многомерный тор», а также подробнее рассматривается маршрутизация сетей Fujitsu Tofu, IBM Blue Gene, Ангара.

Одна из основных проблем, с которой приходится иметь дело при разработке сетей и методов маршрутизации – это отсутствие тупиковых ситуаций, самая часто встречающаяся ситуация носит название «дедлок».

Дедлок (англ. deadlock) – это такое состояние сети, когда группа пакетов не может продолжить движение, поскольку каждому из них для передачи требуется ресурс (например, место в буфере следующего узла), занятый другим. Данная ситуация происходит из-за циклической зависимости при движении сетевого трафика.

Различают статическую и адаптивную маршрутизацию[1]. Статический алгоритм маршрутизации (англ. oblivious routing) – определение маршрута для каждой пары узлов «отправитель»–«получатель» без учета сетевого трафика в сети. Детерминированная маршрутизация – подкласс статической маршрутизации, детерминированный маршрут между двумя заданными узлами всегда один и тот же. Адаптивные алгоритмы, наоборот, учитывают наличие сетевого трафика, маршрут пакета может меняться в зависимости от состояния сети. Данная работа посвящена детерминированной маршрутизации в связи с её первоочередной необходимостью для высокоскоростной сети.

Одномерная маршрутизация является базой для построения многомерной маршрутизации в торе. Каналы, соединяющие соседние узлы, находящиеся в одном измерении тора, образуют кольцо, а в кольце возможны дедлоки. Методы, применяемые для бездедлоковой маршрутизации в кольце: виртуальные каналы и пузырьковая маршрутизация. Виртуальный канал – это пара отдельных выходных и входных буферов для каждого из физических линков, в который маршрутизатор осуществляет отправку и прием сетевых пакетов. Несколько виртуальных каналов позволяют разделить высокоскоростную сеть на несколько виртуальных независимых сетей и таким образом решить проблему дедлоков в кольце. При пузырьковой маршрутизации после прихода нового пакета в кольцо в этом кольце должно оставаться место на еще один пакет максимальной длины, в таком случае движение пакетов в кольце всегда возможно[2].

Многомерная маршрутизация в торе строится над одномерной как следующий уровень. Часто применяется правило порядка измерений или направлений (англ. Direction Order Routing, DOR), который состоит в следующем. Вводится порядок направлений в торе и накладывается ограничение на порядок перемещения пакетов по направлениям. Например, для сети с топологией 3D-тор порядок может быть таким: $+X, +Y, +Z, -X, -Y, -Z$; один пакет может перемещаться по направлениям, скажем, $+X, +Z, -Y$, другой по $+Z, -X, -Y$ и т.п. Каждый пакет перемещается по одному направлению, затем по другому и так далее в заданном порядке до тех пор, пока не достигнет адресата.

[1] Dally W., Towles B. Principles and practices of interconnection networks. Elsevier, 2004. p. 162.

[2] Puente V., Izu C., Beivide R., Gregorio J., Vallejo F., Prellezo J. The adaptive bubble router. Journal of Parallel and Distributed Computing. 61(9), p. 1180–1208, 2001.

Детерминированная маршрутизация может позволять не единственный маршрут, а множество маршрутов между двумя узлами. Часто для выбора конкретного маршрута используется таблица маршрутов, которая может храниться на каждом вычислительном узле. При инжекции пакета в сеть происходит выбор маршрута, по которому детерминированно движется пакет.

Сеть Fujitsu Tofu, Tofu 2/D поддерживает топологию до 6D-тор. Маршрутизация осуществляется по порядку измерений: сначала по измерениям A, B, C, затем по X, Y, Z, затем снова по A, B, C. Ограничения битов направлений нет. Данная маршрутизация позволяет значительные возможности для балансировки нагрузки и обхода отказавших узлов. При совместном использовании суперкомпьютера несколькими заданиями маршруты для детерминированной маршрутизации выбираются таким образом, чтобы не пересекаться между заданиями.

В суперкомпьютерах IBM Blue Gene L/P поддерживается топология 3D-тор, 5D-тор в IBM Blue Gene/Q. Детерминированная маршрутизация устроена в сетях этих суперкомпьютеров одинаково: на одном виртуальном канале в кольце с использованием правила пузырька. Применяется правило порядка измерений, причём с использованием правила битов направлений (dirbit-машрутизация), которое запрещает движение пакета в разных направлениях одного и того же измерения, например, в $+X$ и $-X$ пакету двигаться запрещено.

При совместном использовании суперкомпьютера маршруты пользовательских заданий выбираются таким образом, чтобы не пересекаться между заданиями. При отказе вычислительного узла или канала связи из работы суперкомпьютера исключается 512 вычислительных узлов, имеющих топологию 8x8x8.

Сеть Cray Gemini поддерживает топологию 3D-тор. Каждое направление тора содержит несколько каналов связи. Маршрутизация в сети Gemini основана на правиле порядка измерений. При построении маршрутов используются только кратчайшие, поэтому косвенно можно сделать вывод, что в аппаратно реализованной маршрутизации нет ограничения битов направлений. Трафик различных заданий может пересекаться.

Сеть Ангара поддерживает топологию от 1D до 4D-тор. В маршрутизаторе сети реализована детерминированная маршрутизация, основанная на правиле порядка направлений с использованием битов направлений. Для бездедлоковой маршрутизации в одном измерении (кольце) применяется пузырьковая маршрутизация в одном виртуальном канале. Для расширения возможностей маршрутизации в сети Ангара реализован алгоритм First Step/Last Step (FS/LS) «нестандартного первого и последнего шага», который позволяет сделать первый и последний

шаг без ограничений правила порядка направлений и dirbit-маршрутизации. Например, благодаря FS/LS допустим следующий порядок шагов: $+Y(FS), +X, +Z, -Y, -Z(LS)$.

Детерминированная маршрутизация сети Ангара является расширением детерминированной маршрутизации сетей IBM Blue Gene, так как кроме общих с IBM Blue Gene правил порядка и битов направлений в сети Ангара есть еще дополнительные шаги FS/LS. Для использования FS/LS необходима разработка программных алгоритмов построения таблиц маршрутов, которые будут гарантировать отсутствие дедлоков.

В *разделе 1.3* описываются инструменты, при помощи которых возможно решение задач анализа маршрутов и достижимости сети. Под анализом маршрутов подразумевается набор задач, таких как возможность проверки существования маршрута между двумя узлами сети, построение маршрутов от данного узла до всех остальных в сети.

Самым естественным средством анализа топологии сети и наличия маршрутов в ней является граф сети, в котором вершинами графа являются узлы сети, а ребрами – каналы связи. Однако граф сети неудобен для анализа маршрутов по правилу порядка направлений, так как возможность перехода в ребро из вершины зависит от ребра, по которому произведен переход в данную вершину.

Фундаментальным для всех теорий бездедлоковой маршрутизации в сетях любой топологии является граф зависимостей каналов (англ. channel dependency graph, CDG). *Граф зависимостей каналов* для заданной сети и функции маршрутизации R – это ориентированный граф $G(C, E)$, в котором вершинами являются каналы связи сети, а ребра индуцированы функцией маршрутизации, то есть $(c_p, c_q) \in E$ тогда и только тогда, когда $\exists$ узел сети $n_y : R(c_p, n_y) = c_q$.

Граф зависимостей каналов используется в большинстве работ по анализу достижимости сети и построения маршрутов. Граф зависимостей каналов сети позволяет учитывать маршрутизацию правила порядка направлений в торе, при помощи ограничений на повороты в торе, которые естественным образом отображаются в графе. Например, для канала $+Z$ в некотором узле сети присутствуют ребра переходов в каналы $+Z, -X, -Y, -Z$, но отсутствуют ребра в $+X$ и $+Y$.

Однако маршрутизация в сети Ангара кроме правила порядка направлений ограничена еще требованием dirbit-маршрутизации, поэтому применение графа зависимостей каналов в исходном виде невозможно, так как, например, возможность перехода из $+Z$ в $-X$ в сети Ангара определяется наличием в предыдущих шагах пакета направления $+X$. То же верно и для сетей IBM Blue Gene.

Идея хранить в вершинах графа сети предысторию маршрута используется в статье[3], посвященной построению маршрутизации в гетерогенных

сетях с неустойчивыми связями, например, в сетях, построенных с использованием спутников, мобильных сетей. Для анализа маршрутов в данной статье используется граф, в котором для каждого из N узлов рассматриваемой сети вводятся N вершин графа, которые показывают, из какого предыдущего узла сети маршрут привел в текущий узел. Таким образом, в графе хранится предыстория маршрута, при этом количество вершин графа составляет N^2.

Данная идея с необходимыми изменениями используется в данной диссертационной работе для построения графа с целью анализа маршрутов с учетом всех ограничений маршрутизации сети Ангара.

В *разделе 1.4* рассматриваются алгоритмы построения таблиц маршрутов в высокоскоростных сетях.

Правила маршрутизации, как правило, реализованы аппаратно в высокоскоростных сетях и не могут меняться во время работы системы, также между двумя вычислительными узлами может существовать несколько маршрутов. Для детерминированной маршрутизации возникает задача для каждого узла вычислительной системы построить маршрут до всех остальных узлов, часто для этого используются таблицы маршрутов, которые формируются программно.

Разработано большое количество алгоритмов построения маршрутов в высокоскоростных сетях, отличающихся методами и подходами. Некоторые алгоритмы нацелены на построение таблиц с минимальными маршрутами, часть алгоритмов оптимизирует загрузку сети, другие оптимизируют скорость построения таблиц маршрутов. Также необходимо гарантировать отсутствие дедлоков в сети.

Для решения задачи построения маршрутов популярным методом является применение алгоритмов обработки графов. Например, построение Эйлерова пути или алгоритмы поиска вширь по графу сети или графу зависимостей каналов. Для построения сбалансированных маршрутов кроме алгоритма Дейкстры применяются различные эвристические алгоритмы.

В *разделе 1.5* рассматриваются алгоритмы выбора узлов в суперкомпьютерах. Для сетей с тороидальной топологией существует несколько стратегий выделения ресурсов. Возможно разделение вычислительной системы на партиции, по которым размещаются задачи пользователей. Такая стратегия может снижать эффективность использования кластера из-за выделения большего числа узлов, чем требовалось, или невозможности выделить доступный набор узлов из разных партиций. Данная стратегия использовалась в суперкомпьютерах IBM Blue Gene/P, Blue Gene/Q, в которых ее недостатки компенсировались большим числом не очень мощных

[3]Bulut E., Sahin C., Szymanski B. Conditional shortest path routing in delay tolerant networks. IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks". p. 1–6, 2010.

по производительности вычислительных узлов и адекватным выбором размера партиции.

Вторая стратегия используется в серии суперкомпьютеров Cray XT/XE, где расположение выделенных узлов не зависит от топологии. Такой способ выделения ресурсов может привести к деградации производительности ввиду наличия конкурирующего трафика.

В сети Tofu используется упаковка заданий в торовую топологию, причем эффективность достигается за счет вариативности упаковки в большом количестве измерений (5D). Сетевой трафик заданий пользователей не должен пересекаться.

Существует большое количество алгоритмов для компактного выделения узлов в топологии «многомерный тор». К созданию алгоритмов можно выделить несколько подходов: перебора многомерных прямоугольников, при помощи построения кривой, проходящей через необходимое число узлов, построения многомерных прямоугольников путем расширения из разных узлов.

Перебор многомерных прямоугольников описан для суперкомпьютера IBM Blue Gene/L и заключается в поиске всевозможных многомерных прямоугольников с требуемым числом узлов m, которые можно вписать в топологию. Каждый такой прямоугольник не должен пересекаться с ранее запущенными заданиями. Если m невозможно разложить на множители в торе, то размер искомых прямоугольников увеличивается до ближайшего подходящего.

Основная идея подхода с построением кривой заключается в линейной перенумерации узлов по порядку прохождения некоторой кривой, соединяющей все узлы. Для топологии «многомерный тор» в качестве кривой может использоваться кривая Гильберта. Затем доступные узлы группируются в непрерывные участки в созданной линейной нумерации и выбирается группа с требуемым числом узлов.

В подходе с расширением многомерных прямоугольников из каждого свободного узла строится гиперкуб. Начальный размер гиперкуба равен единице, затем гиперкуб расширяется до тех пор, пока не покроет необходимое число свободных вычислительных узлов.

Примером одновременного решения задачи выделения узлов и отображения узлов задания пользователя на топологию сети может стать алгоритм PaCMap[4]. На этапе отображения на топологию сети граф коммуникаций пользовательского задания с помощью библиотеки METIS разбивается на некоторое число наиболее сильно связанных групп процессов, каждая из которых будет соответствовать одному вычислительному

[4]Tuncer O., Leung V., Coskun A. Pacmap: Topology mapping of unstructured communication patterns onto non-contiguous allocations // Proceedings of the 29th ACM on International Conference on Supercomputing. p. 37–46, 2015.

узлу. Таким образом, задача сводится к выделению необходимого количества узлов из доступных. При этом из полученных групп выделяют центральную, характеризующуюся наименьшей суммой длин кратчайших путей до остальных групп, которая будет отображена на вычислительный узел, обозначенный как n. Остальные группы отображаются на другие узлы итеративно таким образом, чтобы минимизировать использование сети.

На эффективность использования суперкомпьютеров с топологией сети «многомерный тор» влияет фрагментация. Для описания фрагментации в ряде работ вводится понятие прямоугольника максимального размера (ПМР) – это многомерный прямоугольник, который состоит из незанятых узлов и не может быть расширен в любом из направлений тора. Выбор узлов для задания происходит таким образом, чтобы оставшиеся после выделения узлов ПМР были максимального размера.

Вторая глава посвящена описанию предложенного в работе алгоритма построения маршрутного графа для анализа маршрутов в сети с топологией «многомерный тор», на основе которого решается задача определения достижимости исследуемого множества узлов сети.

В *разделе 2.1* вводятся понятия, необходимые для анализа маршрутов, и формулируется постановка задачи. В работе рассматривается *коммуникационная сеть* $I(N, C)$ с топологией «многомерный тор», где N – множество узлов сети, C – множество каналов связи. Размерности тора обозначаются $(d_1, d_2, ..., d_n)$, где n – число измерений тора. Каждый узел сети u имеет координаты $(u_1, u_2, ..., u_n)$, где $0 \leq u_i < d_i$, а также соединен каналами связи с соседними узлами $u + D_j$ в направлениях D_j, где *направление* D_j – это вектор (0,..., $\underbrace{\pm 1}_{j \bmod n}$,...,0) длины n, где на позиции $j \bmod n$ при $1 \leq j \leq n$ будет стоять $+1$ (*положительное направление*) или -1 (*отрицательное направление*) при $n + 1 \leq j \leq 2n$.

Множество направлений обозначается $\mathcal{D} = \{D_j\}_{j=\overline{1,2n}}$. На множестве направлений $\mathcal{D}$ введен порядок в соответствии с нумерацией: $D_i < D_j$, если $i < j$. *Канал связи* (*линк*) можно представить как пару (u, D), где $u \in N$, $D \in \mathcal{D}$, в этом случае множество каналов связи $C = N \times \mathcal{D} \setminus F$, где F – множество отказавших каналов связи.

Определение. *Маршрут P_{u^0,u^l} в сети $I(N,C)$, соединяющий два узла сети u^0 и u^l, – это последовательность вида $u^0, D_1, u^1, D_2, ..., D_l, u^l$, где l – длина маршрута, $u^i \in N, \forall i = \overline{0,l}$, шаг (переход) между узлами u^{j-1} и u^j производится по направлению $D_j, \forall j = \overline{1,l}$, а $(u^j, D_{j+1}) \in C$. При этом $T(P_{u^0,u^l}) = u^1, ..., u^{l-1}$ – транзитные узлы маршрута. Короткая запись маршрута: $u^0, D_1, D_2, ..., D_l$.*

Правила маршрутизации в сети Ангара описываются при помощи следующих определений.

Определение. *Маршрут* $P_{u,v} = u,D_1,D_2,...,D_l$ *из узла* u *в узел* v *сети* I *удовлетворяет правилу порядка направлений, если* $D_{i-1} \leq D_i, \forall i = \overline{2,l}$, *где* l *– длина маршрута,* $D_i \in \mathcal{D}$, $\forall i = \overline{1,l}$.

Определение. *Маршрут* $P_{u,v}^{dirbit} = u^0,S_1,S_2,...,S_{2n}$ *из узла* u *в узел* v *сети* I *удовлетворяет правилу битов направлений, если маршрут* $P_{u,v}^{dirbit}$ *удовлетворяет правилу порядка направлений и* $\forall i = \overline{1,n}$ *наборы шагов* S_i *и* S_{i+n} *в направлениях* D_i *и* $D_{i+n} \in \mathcal{D}$ *удовлетворяют следующему условию: либо* $|S_i| > 0$, *либо* $|S_{i+n}| > 0$, *либо* $|S_{i+n}| = |S_i| = 0$. *Наборы таких направлений* $S_1,S_2,...,S_{2n}$ *обозначаются как* D_{dirbit}.

Определение. *Маршрутизация на основе правила битов направлений в сети* $I(N,C)$ *определяется функцией* $R_{dirbit} : N \times N \to \mathcal{P}(P_{u,v}^{dirbit})$, *где* $\mathcal{P}(P)$ *– множество всех подмножеств* P, *маршрут* $P_{u,v}^{dirbit}$ *удовлетворяет правилу битов направлений.*

Маршрутизация R_{dirbit} при помощи заданного порядка обработки направлений тора обеспечивает отсутствие дедлоков при движении сетевых пакетов между измерениями тора. Для расширения возможностей маршрутизации R_{dirbit} вводится понятие *первого и последнего нестандартного шага* (обозначение – FS/LS), на которые не накладываются ограничения правил порядка и битов направлений.

Определение. *Маршрут* $P_{u,v}^{A}$ *в сети* I *с использованием FS/LS – это маршрут, который может быть представлен как* $u, D_{FS}, D_{dirbit}, D_{LS}$, *где* u *– стартовый узел сети* I, D_{FS} *– положительное направление,* D_{LS} *– отрицательное направление, а* D_{dirbit} *– набор направлений, удовлетворяющих правилу битов направлений. При этом* D_{FS} *и* D_{LS} *могут отсутствовать.*

Определение. *Маршрутизация на основе правила порядка направлений с использованием битов направлений, а также FS/LS в сети* $I(N,C)$ *определяется функцией* $R_A : N \times N \to \mathcal{P}(P_{u,v}^{A})$, *где* $\mathcal{P}(P)$ *– множество всех подмножеств* P, $P_{u,v}^{A}$ *– маршрут с использованием FS/LS.*

В сети Ангара аппаратно реализована маршрутизация R_A на основе правила порядка направлений с использованием битов направлений, а также FS/LS. Заметим, что такая маршрутизация может приводить к дедлокам в сети, так как на FS/LS не накладывается требование порядка направлений.

В диссертационной работе поставлено требование отсутствия пересечения сетевого трафика для разных задач, запускаемых на вычислительной системе. Это ограничение порождает задачу определения достижимости множества вычислительных узлов в заданной сети.

Определение. *Множество узлов* $M_{a,t}$ *сети* $I(N,C)$ *достижимо, если* $\forall u, v \in M_a, u \neq v \; \exists P_{u,v} : T(P_{u,v}) \subset M_{a,t}$, *где* $M_{a,t} = M_a \cup M_t$, *где* $M_a \subseteq N$, $M_t \subset N$ *– множества активных и транзитных узлов.*

Определение. *Пусть даны сеть* $I(N,C)$, *а также функция маршрутизации* $R : N \times N \to \mathcal{P}(P_{u,v})$, *где* $\mathcal{P}(P)$ *– множество всех подмножеств* P,

$P^R_{u,v}$ – маршрут, соединяющий два узла сети u и v согласно функции R. Маршрутным графом сети I с функцией маршрутизации R называется граф $RG(V,E)$, в котором маршрут $P^R_{u,v}$ в сети I между ее узлами u,v существует тогда и только тогда, когда в маршрутном графе существует путь между вершинами, соответствующими узлам u и v.

Задача разработки алгоритма анализа маршрутов в сети Ангара формулируется следующим образом. Пусть R_{A^-} – это маршрутизация R_A, в которой на нестандартные шаги FS/LS накладывается требование правила порядка направлений. Так как маршрутизация R_A допускает дедлоки, то необходимо построить бездедлоковую функцию маршрутизации $R_{A^*} \supseteq R_{A^-}$. Для сети I требуется построить маршрутный граф для функции маршрутизации R_{A^*}.

Раздел 2.2 посвящен разработке алгоритма построения маршрутного графа для маршрутизации R_{A^-}, при которой дедлоки в сети невозможны.

Рассмотрим ориентированный граф $RG(V,E)$. Вершины графа будем обозначать U^i_X. Каждому вычислительному узлу сети $I(N,C)$ соответствует несколько вершин графа. Верхний индекс i определяет, какому узлу сети соответствует данная вершина. Нижний индекс X определяет информацию о предыстории маршрута, которая в соответствии с правилами маршрутизации вносит ограничения на принятие решения о следующем шаге.

Множество U^i_X состоит из следующих вершин:

1. U^i_{begin} – вершина, из которой начинается движение (инжекция пакета в сеть);
2. $U^i_{FS_j}, j = \overline{1,n}$ – вершины, в которые можно попасть, совершив первый нестандартный шаг из соседнего узла в узел i в положительном направлении D_j;
3. $U^i_{Dirbit_j}, j = \overline{1,3^n-1}$, где индекс $Dirbit_j$ – набор направлений, удовлетворяющих правилу битов направлений, двигаясь по которым можно попасть в узел i;
4. $U^i_{LS_j}, j = \overline{n+1,2n}$ – вершины, в которые можно попасть, совершив последний нестандартный шаг из соседнего узла в узел i отрицательном направлении D_j;
5. U^i_{end} – вершина, в которой заканчивается движение (эжекция пакета из сети).

Ребра маршрутного графа строятся так, что переход от одной вершины графа к другой соответствует маршруту, проходящему через соответствующие узлы сети I и удовлетворяющему маршрутизации R_{A^-}.

В работе сформулирована и доказана следующая теорема:

Теорема. *Из узла a в узел b сети I существует маршрут $P_{a,b}$, удовлетворяющий маршрутизации R_{A^-}, тогда и только тогда, когда в графе $RG(V,E)$ существует путь из вершины U^a_{begin} в вершину U^b_{end}.*

По доказанной теореме построенный граф $RG(V,E)$ является маршрутным. Граф $RG(V,E)$ содержит $|N| * (3^n + 2n + 1)$ вершин и

$|N| * (2n3^n + 1.5n^2 + 1.5n + 1)$ ребер, где $|N|$ – число узлов в сети, n – размерность тора.

В *разделе 2.3* рассматривается случай, когда нестандартные шаги FS/LS могут нарушать правило порядка направлений, при этом в сети возможны дедлоки.

Сначала описывается построение графа зависимостей каналов для сети с маршрутизацией с правилом порядка направлений.

Определение. *Маршрутизация с правилом порядка направлений в сети* $I(N, C)$ *определяется функцией маршрутизации* $R_{DOR} : C \times N \to \mathcal{P}(C)$, $\mathcal{P}(C)$ *– множество всех подмножеств* C. *Функция* R_{DOR} *для канала связи* $c_{q-1} \in C$ *на текущем шаге маршрута* $P_{*,n}$ *в узел-получатель* n *возвращает набор возможных каналов связи* $\{c_q^1, ..., c_q^p\}$ *следующего шага маршрута* $P_{*,n}$, *причем должно выполняться условие* $D(c_{q-1}) \leq D(c_q^i), \forall i = \overline{1,p}$, *где* $D(c)$ *– направление тора для канала связи* c.

Рассмотрим граф зависимости каналов $G(C, E)$, индуцированный маршрутизацией R_{DOR}. Обозначим $D(P_{u,v})$ – набор направлений тора, участвующих в маршруте $P_{u,v}$, где $u, v \in N$.

Определение. *Используемым набором направлений* $B(u_i, D_i)$ *в сети* $I(N,C)$ *для заданных узла сети* u_i *и направления* D_i *называется множество, состоящее из объединения наборов направлений тора* $D(P_{u,v})$ *всех маршрутов* $P_{u,v}$, *проходящих через узел* u_i *и совершающих шаг из этого узла в направлении* D_i. *То есть* $B(u_i, D_i) = \bigcup D(P_{u,v})$, $\forall u,v \in N$: *маршрут* $P_{u,v}$ *содержит шаг из узла* u_i *в направлении* D_i.

В работе сформулирована и доказана следующая теорема.

Теорема. *Пусть граф зависимостей каналов* $G(C, E)$ *построен для сети* $I(N, C)$ *и маршрутизации* R_{DOR}. *Дополнительное ребро* $e = ((u_i,D_i),(u_j,D_j))$, *где* $e \notin E, u_i + D_i = u_j$, *не создаст дополнительных циклов в* G, *если* $D_i \notin B(u_j, D_j)$.

В работе предложен алгоритм построения дополнительных ребер, который в графе зависимостей каналов $G(C, E)$ для маршрутизации R_{DOR} добавляет по одному ребру, не создающих новых циклов. На основе доказанной ранее теоремы[5] о бездедлоковости маршрутизации, если в соответствующем графе зависимости каналов нет циклов, в работе доказывается следующая теорема:

Теорема. *Маршрутизация* R_{DOR^*}, *индуцированная модифицированным графом зависимостей каналов при помощи алгоритма построения дополнительных ребер, является бездедлоковой.*

Затем осуществляется переход от анализа графа зависимости каналов, который не позволяет учитывать ограничения правила битов

[5]Dally W., Seitz C. Deadlock-free message routing in multiprocessor interconnection networks. IEEE Transactions on computers. 100(5), p. 547–553, 1988.

направлений маршрутизации R_{dirbit}, к построению маршрутного графа, который позволяет учитывать данное ограничение.

В диссертационной работе разработан алгоритм *RoutingGraphAdd*, который для каждого построенного дополнительного ребра в графе зависимостей каналов добавляет в построенный в предыдущем разделе маршрутный граф дополнительные ребра для вершин, соответствующих FS/LS.

Определение. *Маршрутизация R_{A^*} с нарушением правила порядка направлений для FS/LS для сети I определяется функцией, которая индуцирована маршрутным графом с добавленными ребрами при помощи алгоритма RoutingGraphAdd.*

В данной работе доказаны следующие теоремы о построенной маршрутизации R_{A^*}.

Теорема. *Маршрутизация R_{A^*} является бездедлоковой.*

Теорема. *Маршрут $P_{u,v}$ из узла u в узел v сети I, удовлетворяющий маршрутизации R_{A^*}, существует тогда и только тогда, когда в графе $RG(V,E)$, соответствующем маршрутизации R_{A^*}, существует путь из вершины U^a_{begin} в вершину U^b_{end}.*

По доказанной теореме построенный при помощи алгоритма *RoutingGraphAdd* граф $RG(V,E)$ является маршрутным. Временная сложность алгоритма построения маршрутного графа составляет $O(|N|^2)$, где $|N|$ – количество узлов сети.

В *разделе 2.4* формулируется задача определения достижимости множества узлов и описывается предложенный в работе алгоритм её решения. Для сети I и заданной функции маршрутизации R_{A^*} требуется разработать алгоритм определения достижимости множества узлов $M_{a,t}$.

Алгоритм основан на возможности сведения данной задачи к проверке связности вершин в соответствующем сети I и маршрутизации R_{A^*} маршрутном графе RG. В графе RG удаляются все ребра, ведущие из вершин для узлов $M_{a,t}$ к вершинам других узлов сети. Для каждой из вершин U^i_{begin} графа, соответствующих узлам сети из множества M_a, при помощи поиска вширь в графе определяется множество достижимых из нее конечных вершин вида U^j_{end}, данное множество должно содержать вершины для всех узлов множества M_a. При этом транзитные узлы должны принадлежать $M_{a,t}$. Сложность алгоритма оценивается как $O(|M_{a,t}| \cdot |M_a|)$.

Необходимо отметить, что предложенный маршрутный граф, а также алгоритм определения достижимости множества узлов применимы не только для сети Ангара, но и для сетей серии IBM Blue Gene.

В третьей главе рассматривается проблема построения таблицы маршрутов для решения задачи балансировки трафика. В начале главы вводятся необходимые определения и формулируется постановка задачи.

Определение. *Таблицей маршрутов RT достижимого множества узлов* $M_{a,t} = M_a \cup M_t$ *сети I назовем множество маршрутов* $P_{u,v}$ *таких, что для любых двух активных узлов* $u, v \in M_a \; \exists \,!\; P_{u,v} \in RT$.

Так как маршрутов между каждой парой вычислительных узлов может быть несколько, необходимы критерии оценки предложенной таблицы маршрутов. Для этого в работе используются следующие характеристики таблицы маршрутов.

Определение. *Диаметром достижимого множества узлов* $M_{a,t}$ *сети с таблицей маршрутов RT назовем максимальную длину маршрута в RT.*

Определение. *Загруженностью* G_c *канала связи с для таблицы маршрутов RT в сети I будем называть количество маршрутов* $P_{i,j}$, *которым принадлежит данный канал связи:* $G_c = \mid \{P_{i,j} : c \in P_{i,j}, P_{i,j} \in RT\} \mid$.

Определение. *Множеством каналов связи* $C(M_{a,t})$ *множества узлов* $M_{a,t}$ *сети* $I(N,C)$ *назовем все такие* $(u,D) \in C, u \in M_{a,t}$, *что* $\exists v \in M_{a,t} : v = u + D$.

Определение. *Идеальной загруженностью канала связи* $\pi_{perfect}$ *множества узлов* $M_{a,t}$ *сети I назовем среднюю загруженность каналов связи сети для любой таблицы маршрутов RT для* $M_{a,t}$, *содержащей только маршруты минимальной длины:* $\pi_{perfect} = \frac{\sum_{c \in C(M_{a,t})} G_c}{|C(M_{a,t})|}$.

Определение. *Максимальной загруженностью канала связи для таблицы маршрутов RT достижимого множества узлов* $M_{a,t}$ *сети I назовем* $\pi_{max}(RT) = \max\limits_{c \in C(M_{a,t})} G_c$.

Даже при небольшом числе узлов сети и вариантов маршрутов между узлами число различных таблиц маршрутов очень велико, поэтому с точки зрения практики достаточно получить приближенное решение задачи балансировки трафика. Задача построения таблиц маршрутов для приближенного решения задачи балансировки трафика формулируется следующим образом. Для достижимого множества узлов $M_{a,t}$ сети I с заданной функцией маршрутизации R_{A^*} требуется разработать алгоритм построения сбалансированных таблиц маршрутов RT, при этом необходимо оценить пригодность алгоритма с точки зрения применения на практике: 1) по времени выполнения алгоритма; 2) по критерию оценки сбалансированности – минимизация $\pi_{max}(RT)$.

В *разделе 3.1* описывается разработанный ранее базовый алгоритм построения таблиц маршрутов, в котором распределение сетевого трафика зависит от координат начального узла. К недостаткам базового алгоритма относится невозможность работы с множествами узлов сложной конфигурации и невозможность учета сломанных или недоступных ресурсов.

В *разделах 3.2* и *3.3* описаны предложенные алгоритмы построения таблиц маршрутов BFS и генетический алгоритм. В обоих алгоритмах по заданному количеству транзитных узлов их множество выбирается случайно внутри множества $M_{a,t}$.

В алгоритме BFS изначально все каналы связи множества узлов $M_{a,t}$ сети I имеют нулевую загруженность. Для каждого активного узла $u \in M_a$ в графе $RG(V, E)$ запускается алгоритм поиска вширь. После окончания поиска из каждого узла множества M_a необходимо подняться по построенному дереву обратно вверх к узлу u, увеличивая при этом загруженность G_i проходимых каналов связи сети. Для получения более сбалансированных таблиц маршрутов применено две эвристики. В первой следующий узел сети для запуска поиска вширь выбирается максимально удаленным от предыдущего узла. Во второй выполняется сортировка вершин на каждом новом слое поиска вширь по возрастанию загруженности каналов связи, соответствующих вершинам. Предложенный алгоритм строит таблицу маршрутов минимальной длины. Сложность алгоритма оценивается как $O(|M_{a,t}| \cdot |M_a|)$.

В генетическом алгоритме за ген взят маршрут между двумя узлами сети I, индивидуум – таблица маршрутов (набор генов). Определены операции выбора родителя, скрещивания, мутации и вычисления пригодности индивидуума. Сложность одной интерации этого алгоритма – $O(k|M_a|^2)$, где k – количество индивидуумов в популяции.

В четвертой главе рассматривается задача выбора множества вычислительных узлов требуемого размера.

Определение. *Состоянием сети $I(N, C)$ будем называть пару множеств (S_N, S_C), где $S_N \subseteq N$ – множество недоступных узлов, $S_C \subseteq C$ – множество недоступных каналов связи.*

Задача выбора узлов формулируется следующим образом. Для сети $I(N,C)$ и ее состояния (S_N, S_C), функции маршрутизации R_{A^*}, чисел m и T активных и транзитных узлов требуется разработать алгоритм выбора достижимого множества вычислительных узлов $M_{a,t}$, такого что $|M_a| = m$, $|M_{a,t}| \leq m + T$, $M_{a,t} \cap S_N = \varnothing$. Для выбора единственного решения из набора возможных вводится упорядоченный набор критериев оценки множества узлов:

1. минимизация числа транзитных узлов;
2. минимизация фрагментированности сети после выделения множества узлов;
3. минимизация диаметра таблицы маршрутов;
4. минимизация оценки таблиц маршрутов π_{max}.

Первый и второй критерии самые важные, так как необходимы для эффективного использования аппаратных ресурсов вычислительного кластера. Третий критерий позволяет минимизировать максимальную задержку передачи данных. Последний критерий максимизирует сбалансированность получаемой таблицы маршрутов.

В *разделе 4.1* описывается способ оценки фрагментированности сети после выделения множества узлов. *Прямоугольник максимально возможного размера (ПМР)* в сети $I(N, C)$ с состоянием (S_N, S_C) – многомерный

прямоугольник, состоящий только из доступных узлов (т.е. из узлов множества $N \setminus S_N$), который нельзя расширить ни в одну из его сторон. Расширить прямоугольник может быть невозможно по двум причинам: либо по соответствующему измерению тора достигнуто максимальное количество узлов в кольце (расширять некуда), либо сторона прямоугольника граничит с недоступным узлом. Множество различных ПМР характеризуют меру фрагментированности сети. В диссертационной работе предложен алгоритм поиска таких прямоугольников.

В качестве критерия оценки сети $I(N, C)$ с состоянием $S = (S_N, S_C)$ на основе прямоугольников максимального размера вводится функция $\varphi(I, S)$ от числа найденных ПМР, которая тем больше, чем большее число прямоугольников максимального размера имеется в сети: $\varphi(I, S) = |N| * MSS_{max}^{nnodes} + |MSS_{max}|$, где MSS_{max}^{nnodes} – количество узлов в ПМР максимального размера, $|MSS_{max}|$ – число таких ПМР.

Для каждого найденного достижимого множества $M_{a,t}$ оценивается значение функции $\varphi(I, S')$, где S' – состояние сети после предполагаемого выделения узлов, т.е. $S' = (S_N \cup M_{a,t}, S_C \cup C(M_{a,t}))$. Для увеличения утилизации сети требуется выбирать решения с наибольшим значением функции φ.

В *разделе 4.2* описывается общий алгоритм перебора n-мерных прямоугольников. На вход алгоритму подается сеть I, ее состояние S, требуемое число узлов m, максимально допустимое число транзитных узлов T. Алгоритм перебора n-мерных прямоугольников ищет всевозможные прямоугольники допустимых размеров, которые можно вписать в рассматриваемую сеть. Любой прямоугольник можно описать двумя параметрами: размер прямоугольника $P = (p_1,...,p_n)$ и его расположение в сети. Перебор всевозможных n-мерных прямоугольников можно представить как $m \leq |P| = \prod_{i=1}^{n} p_i \leq m + T$, $\forall i = \overline{1,n}$ $0 < p_i \leq d_i$. Сложность такого перебора $O(\prod_{i=1}^{n} d_i) = O(N)$, где N – число узлов сети. Для каждого найденного прямоугольника подбирается его расположение в сети, в худшем случае таких расположений N.

В *разделе 4.3* разделе описывается разработанный ранее базовый алгоритм выбора множества узлов перебором n-мерных прямоугольников. Базовый алгоритм выбора узлов обладает следующими недостатками:

1. В алгоритме общий алгоритм перебора n-мерных прямоугольников ограничен условием $1 \leq p_i \leq \lceil d_i \rceil$ `or` $p_i = d_i$ для того, чтобы возможно было применить базовый алгоритм построения таблиц маршрутов. Также нельзя рассматривать прямоугольник с недоступными ресурсами, хотя он может быть достижимым. Из-за этого происходит потеря возможных решений задачи выбора узлов.
2. Отсутствует возможность выделения для задачи множества узлов, по форме отличного от многомерного прямоугольника.

В *разделе 4.4* описывается предложенный улучшенный алгоритм выбора множества узлов перебором n-мерных прямоугольников, который основан на общем алгоритме перебора. Для каждого n-мерного прямоугольника проверяется достижимость узлов и строится таблица маршрутов. Транзитные и активные узлы выбираются случайным образом. В результате работы алгоритма получается набор достижимых множеств с заданным числом активных узлов. Временная сложность алгоритма оценивается как $O((m+T)\cdot m\cdot N^2)$, где m – требуемое число узлов, T – допустимое число транзитных узлов, N – число узлов сети.

Разработанный алгоритм снимает недостатки базового алгоритма, которые становятся его достоинствами:

1. В алгоритме снимается ограничение на перебор прямоугольников, в результате появляется возможность найти новые возможные решения задачи.
2. Присутствует возможность выделения для задания множества узлов, по форме отличного от многомерного прямоугольника.

Результатом работы алгоритма выбора множества узлов в сети I с состоянием (S_N, S_C) является набор допустимых решений, из которых согласно введенным критериям выбирается одно решение.

В пятой главе приведены исследования отказоустойчивости различных сетей на основе разработанного алгоритма определения достижимости множества вычислительных узлов, алгоритмов построения сбалансированной таблицы маршрутов и выбора узлов, а также общей эффективности работы вычислительного кластера.

В разделах 5.1 и 5.2 исследования проводятся для всевозможных сетей с числом узлов $N = \prod_{i=1}^{n} d_i \leq 128, d_i \leq 8, 2 \leq n \leq 4$, где d_i – размер i-го измерения, n – размерность тора. Число узлов 128 превышает количество узлов в самом большом суперкомпьютере на основе сети Ангара, поэтому указанные сети охватывают необходимые практические случаи.

В *разделе 5.1* приводится исследование отказоустойчивости сети с использованием разработанного алгоритма определения достижимости множества узлов. В среднем возможность нарушения правила порядка направлений в маршрутизации R_{A^*} по сравнению с маршрутизацией R_{A^-} без возможности нарушения данного правила позволяет увеличить число случайно сломанных каналов связи до потери достижимости на 4,9%, 8,2% и 34% для сетей с топологией 2D, 3D и 4D-тор, соответственно.

Раздел 5.2 посвящен сравнению времени работы разработанных алгоритмов построения таблиц маршрутов и качества получаемых таблиц маршрутов. Качество построенных таблиц маршрутов оценивалось с помощью *коэффициента сбалансированности* $balance\ factor = (\frac{\pi_{max}(RT)}{\pi_{perfect}} - 1) * 100\%$. Коэффициент сбалансированности характеризует отклонение приближенного решения задачи балансировки сетевого трафика от оптимального решения.

В исследовании качества получаемых таблиц маршрутов рассматривались всевозможные сети с числом узлов до 128, таблицы маршрутов строились для всех узлов каждой сети, все узлы считались активными. Для сетей без отказов возможно сравнение разработанных алгоритмов с базовым алгоритмом построения таблиц маршрутов, который изначально существовал в системном программном обеспечении сети Ангара. Средний (максимальный) проигрыш коэффициента сбалансированности для алгоритма BFS базовому алгоритму составляет 13% (30%), а по сравнению с генетическим алгоритмом – 32% (36%).

В исследовании по времени работы разработанных алгоритмов за ограничение сверху на построение таблицы маршрутов и выбор узлов взято отведенное в системе управления заданиями Slurm время на работу плагина по выделению узлов. Для Slurm по умолчанию оно составляет 10 секунд. Время работы генетического алгоритма построения таблицы маршрутов для сетей с количеством узлов больше 48 становится недопустимо большим. Время работы алгоритма BFS на сетях с числом узлов до 1024 показывает допустимые результаты, поэтому в дальнейших исследованиях используется именно этот алгоритм.

В *разделе 5.3* описывается разработанная имитационная модель вычислительной системы с заданной сетью. На вход имитационной модели подается сеть и поток пользовательских заданий. Каждое задание i характеризуется требуемым числом узлов сети m_i и временем W^i_{time}, необходимым для задания. На выходе выдается полное время работы всего кластера T, полезное время работы каждого узла сети T_i и время начала предоставления ресурсов для каждого задания. Используя эти данные, можно вычислить утилизацию ресурсов сети U и относительное время ожидания задания в очереди T_{mean}.

Под *утилизацией* ресурсов U сети понимается среднее значение утилизации по всем узлам сети: $U = \frac{\sum_{i=1}^{|N|} U_i}{N}, U_i = \frac{T_i}{T}$, где U_i – утилизация i-го узла сети, T – полное время работы вычислительного кластера, T_i – полезное время работы i-го узла сети, N – общее количество узлов сети. *Относительное время ожидания задания в очереди* $T_{mean} = \frac{\sum_{i=1}^{k} T^i_{delay}}{k} = \frac{1}{k}\sum_{i=1}^{k} \frac{Q^i}{W^i_{time}}$, где k – число различных заданий в потоке, W^i_{time} – запрошенное время для задания i, Q^i – время ожидания задания i в очереди.

Модель поочередно выбирает задания из очереди заданий в рамках некоторого окна размера w и пытается выделить ресурсы. Узлы в сети выделяются на требуемое заданием время. Работа модели заканчивается, когда все задания из очереди будут выполнены. Время поиска заданий не учитывается, так как оно может быть выполнено на фоне работы заданий.

В диссертационной работе рассматриваются два типа очередей заданий: очередь заданий с реального суперкомпьютера «Десмос» (ОИВТ

РАН) из 32 узлов с сетью Ангара; синтетические очереди заданий, созданные по правилам, приближенным к реальным условиям. Максимальное время работы задания выбрано равным 1 суткам. Очередь создавалась таким образом, чтобы обеспечить теоретическую возможную загрузку кластера 80%. Все задания случайно распределялись по временной шкале в диапазоне 1 месяца для очереди суперкомпьютера «Десмос» и 4 месяца для остальных очередей.

В *разделе 5.4* приводятся результаты исследования сравнения количества возможных решений, найденных с помощью разработанного алгоритма выбора узлов улучшенным перебором многомерных прямоугольников и с помощью базового алгоритма выбора узлов, который изначально работал на вычислительных кластерах с сетью Ангара.

В исследовании рассматривались различные сети с числом узлов от 32 до 256. Для каждой сети запускалась имитационная модель с окном заданий размера 1 и соответствующей синтетической очередью заданий. Модель обрабатывала очередь с помощью базового алгоритма, при этом запоминалось количество найденных решений для каждого запуска базового алгоритма. Перед каждым запуском базового алгоритма также запускался разработанный улучшенный алгоритм выбора узлов, для которого также запоминалось количество найденных решений. Описанные условия приближают сравнение к реальным системам, а представленный набор сетей шире набора топологий существующих суперкомпьютеров на основе сети Ангара. В таблице 1 представлено среднее значение найденных решений для обоих алгоритмов для каждой сети и их отношение. Разработанный алгоритм выбора узлов предлагает в 2–12 раз больше вариантов решений, чем базовый.

В *разделе 5.5* проводится исследование утилизации сетей с использованием разработанных алгоритмов. Сравнение разработанных алгоритмов выбора узлов и построения таблиц маршрутов проводилось с базовыми алгоритмами, которые изначально работали на вычислительных кластерах с сетью Ангара.

Раздел 5.5.1 посвящен сравнению результатов на реальном суперкомпьютере «Десмос» с результатами, полученными на разработанной имитационной модели вычислительного кластера.

В таблице 2 представлены результаты выполнения одной и той же очереди заданий на суперкомпьютере «Десмос» и на имитационной модели при работе разработанного и базового алгоритма выбора узлов. Адекватность имитационной модели подтверждается разницей полученной утилизации на модели и суперкомпьютере – не более 1,2% абсолютных пунктов. Улучшение утилизации суперкомпьютера при применении разработанного алгоритма выбора узлов по сравнению с базовым составляет 7,65%. Аналогично улучшилась и оценка времени ожидания задания в очереди, которое сократилось для суперкомпьютера в 2,12 раз.

Таблица 1 — Исследование среднего количества найденных решений, найденных базовым и разработанным улучшенным алгоритмами выбора узлов на имитационной модели.

Сеть	Кол-во узлов	Среднее число найденных решений		Отношение разработанного алг. к базовому
		Базовый алгоритм	Разработанный алгоритм	
4x4x2	32	8,29	28,54	3,44
4x2x2x2	32	8,26	26,08	3,16
4x3x3	36	11,41	29,20	2,56
3x3x2x2	36	11,49	22,79	1,98
4x4x4	64	15,28	92,85	6,08
4x4x2x2	64	15,27	75,15	4,92
6x4x4	96	39,18	237,54	6,06
4x4x3x2	96	32,69	166,15	5,08
8x6x3	144	95,08	479,96	5,05
4x4x3x3	144	98,93	520,61	5,26
8x8x4	256	214,93	1179,75	5,49
4x4x4x4	256	162,29	1929,15	11,89

Таблица 2 — Результаты выполнения очереди заданий на суперкомпьютере «Десмос» и на имитационной модели при работе разработанного и базового алгоритмов выбора узлов.

Характеристика	Базовый алгоритм		Разраб. алгоритм		Сравнение	
	Мод.	Десмос	Мод.	Десмос	Мод.	Десмос
Утилизация U	58,54%	59,35%	65,80%	67,00%	7,26%	7,65%
Ожидание T_{mean}	17,02	18,28	7,34	8,63	2,32	2,12

В *разделах 5.5.3 и 5.5.4* проведено исследование утилизации и относительного времени ожидания задания в очереди на сетях от 32 до 1024 узлов с топологиями 3D и 4D-тор от 4x4x2 до 8x8x4x4. Данный набор сетей охватывает как практические случаи, так и дает возможность оценить применимость алгоритмов для суперкомпьютеров с большим числом узлов.

Использование в имитационной модели возможности выбрать очередное задание в рамках окна заданий размером от 1 до 128 соответствует применению более сложных по сравнению с FIFO политик планирования в Slurm. В данном режиме разработанный алгоритм выбора узлов с применением разработанной оценки фрагментации в среднем по всем сетям позволяет достичь увеличения утилизации вычислительного кластера (минимум – максимум) на 6,96% (0,5%–16,5%) относительно базового алгоритма выбора узлов. При этом уменьшение относительного времени ожидания задания в очереди составило 1,83 раз по сравнению с базовым алгоритмом. С ростом размера окна утилизация во всех экспериментах

увеличивается, а относительное время ожидания задания в очереди сокращается.

На рисунке 1 представлено время работы алгоритма выбора узлов для поиска одного решения, усредненное по всем заданиям в потоке пользовательских заданий. Время работы разработанных алгоритмов является допустимым для эксплуатации на реальных вычислительных системах.

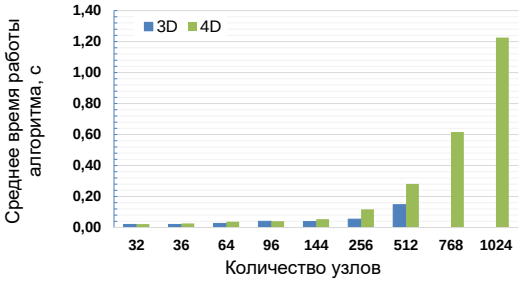

Рис. 1 — Среднее время работы алгоритма выбора узлов в зависимости от количества узлов в сети.

В заключении диссертационной работы перечисляются ее основные результаты, которые заключаются в следующем:

1. Разработан алгоритм построения маршрутного графа для анализа маршрутов в высокоскоростных коммуникационных сетях с топологией «многомерный тор» с произвольным количеством отказавших узлов и каналов связи, а также маршрутизацией, накладывающей ограничение на маршрут сетевого пакета в зависимости от истории его прохождения по сети. Временная сложность алгоритма $O(N^2)$, где N – количество узлов в сети.
2. Разработан алгоритм определения достижимости множества вычислительных узлов сети размера N, временная сложность алгоритма $O(N^2)$. Алгоритм использует возможность программного контроля отсутствия дедлоков в сети Ангара, что позволяет сохранять достижимость сети при большем числе случайно отказавших каналов связи (от 5% до 34%) по сравнению с возложением контроля отсутствия дедлоков на аппаратные возможности сети Ангара.
3. Разработан алгоритм построения таблицы маршрутов для решения задачи балансировки трафика в достижимом множестве узлов размера N, временная сложность алгоритма $O(N^2)$.
4. Разработан алгоритм выбора узлов в сети размера N с учетом её фрагментации, временная сложность алгоритма $O(N^4)$. Алгоритм позволил по сравнению с существовавшими ранее алгоритмами от 2 до 12 раз расширить возможности при выборе множества узлов в сети Ангара в зависимости от потока пользовательских заданий и исследуемой системы.
5. Проведено экспериментальное исследование разработанных алгоритмов, которое по сравнению с базовыми алгоритмами показало, в среднем, улучшение утилизации вычислительных систем на 7%, а относительного времени ожидания задания в очереди – в 1,83 раза.

Публикации по теме диссертации

Публикации в изданиях, индексируемых в базах данных Web of Science, Scopus и RSCI

1. *Mukosey Anatoly, Semenov Alexander, Tretiakov Aleksandr*. Graph Based Routing Algorithm for Torus Topology and Its Evaluation for the Angara Interconnect // *Journal of Parallel and Distributed Computing*. — 2024. — Vol. 183. — P. 104765. — [WoS: Impact Factor 3.7].

2. *Mukosey Anatoly, Semenov Alexander*. Simulation of Utilization and Energy Saving of the Angara Interconnect // *Lobachevskii Journal of Mathematics*. — 2022. — Vol. 43, no. 4. — Pp. 873–881. — [WoS: Impact Factor 0.7].

3. *Mukosey Anatoly, Simonov Alexey, Semenov Alexander*. Extended Routing Table Generation Algorithm for the Angara Interconnect // Supercomputing: 5th Russian Supercomputing Days, RuSCDays 2019. — Vol. 1129. — Springer, 2019. — Pp. 573–583. — [Scopus: Impact Factor 0.49].

4. Early Performance Evaluation of the Hybrid Cluster with Torus Interconnect Aimed at Molecular-Dynamics Simulations / Vladimir Stegailov, Alexander Agarkov, Anatoly Mukosey et al. // International Conference on Parallel Processing and Applied Mathematics. — Springer, 2017. — Pp. 327–336. — [Scopus: Impact Factor 0.969].

Публикации в журналах, входящих в перечень изданий, рекомендованных ВАК при Минобрнауки России

5. *Мукосей А.В., Семенов А.С.* Оптимизация фрагментации при выделении ресурсов для высокопроизводительных вычислительных систем с сетью Ангара // *Вестник ЮУрГУ, серия Вычислительная математика и информатика*. — 2018. — Т. 7, № 2. — С. 50–62. — [РИНЦ: импакт-фактор 0.524].

6. *Мукосей А.В., Симонов А.С., Семенов А.С.* Оптимизация утилизации при выделении ресурсов для высокопроизводительных вычислительных систем с сетью Ангара // *Вестник ЮУрГУ, серия Вычислительная математика и информатика*. — 2019. — Т. 8, № 1. — С. 5–19. — [РИНЦ: импакт-фактор 0.524].

7. *Мукосей А.В., Семенов А.С.* Приближенный алгоритм выбора оптимального подмножества узлов в коммуникационной сети Ангара с отказами // *Вычислительные методы и программирование*. — 2017. — Т. 18, № 1. — С. 53–64. — [РИНЦ: импакт-фактор 0.576].

Иные публикации

8. *Мукосей А.В., Семенов А.С.* Оптимизация фрагментации при выделении ресурсов для высокопроизводительных вычислительных систем с сетью Ангара. // Параллельные вычислительные технологии (ПаВТ'2018): труды международной научной конференции. — Издательский центр ЮУрГУ, 2018. — С. 310–318.

9. *Мукосей А.В., Семенов А.С.* Оптимизация утилизации при выделении ресурсов для высокопроизводительных вычислительных систем с сетью Ангара // Суперкомпьютерные дни в России 2018: Труды международной конференции. — М.: Изд-во МГУ, 2018. — С. 831–840.

10. *Мукосей А.В., Семенов А.С., Макагон Д.В.* Приближенный алгоритм выбора оптимального подмножества узлов в коммуникационной сети Ангара с отказами // Параллельные вычислительные технологии (ПаВТ'2016): труды международной научной конференции. — Издательский центр ЮУрГУ, 2016. — С. 257–269.

11. *Мукосей А.В.* Приближенное решение задачи оптимального распределения трафика в коммуникационной сети с топологией «многомерный тор» // Суперкомпьютерные дни в России: Труды международной конференции. — М.: Изд-во МГУ, 2016. — С. 864–874.

Мукосей Анатолий Викторович

Алгоритмы выбора узлов и построения таблиц маршрутов для высокоскоростной сети с топологией «многомерный тор»

Автореф. дис. на соискание ученой степени к.ф.-м.н.

Подписано в печать ___.___._____. Заказ № _____
Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.
Типография _____